

LEARNING C

ABHISHEK DWIVEDI
IT DEPARTMENT
AVIT, CHENNAI.

WHICH TOPIC U WANNA STUDY ???

INTRODUCTION

CONSTANTS & VARIABLES

OPERATORS & EXPRESSIONS

STRUCTURE OF A C PROGRAM

CONTROL STRUCTURES

ARRAYS AND STRINGS

FUNCTIONS

STORAGE CLASSES

STRUCTURES & UNIONS

POINTERS

DYNAMIC MEMORY ALLOCATION

FILE MANAGEMENT IN C

COMMAND LINE ARGUMENTS



Introducing C

- C is a programming language developed at AT & T Bell Laboratories of USA in 1972, designed and written by "Dennis Ritchie".
- C is highly portable i.e., software written for one computer can be run on another computer.
- An important feature of 'C' is its ability to extend itself. A C program is basically a collection of functions.



Historical Development of C

Year	Language	Developed by	Remarks
1960	ALGOL	International Committee	Too general, too abstract
1963	CPL	Cambridge University	Hard to learn, difficult to implement
1967	BCPL	Martin Richards at Cambridge University	Could deal with only specific problems
1970	B	Ken Thomson at AT & T	Could deal with only specific problems
1972	C	Dennis Ritchie at AT & T	Lost generality of BCPL and B restored

ALGOL



Algorithmic Language

CPL



Combined Programming Language

BCPL



Basic Combined Programming Language





C Tokens

- A token is an atomic unit (smallest indivisible units) in a program.
- The most basic elements in a C program recognized by the compiler are a single character or a group of characters called C tokens.
- The compiler cannot breakdown the token any further. For example, the words main, '{' (brace), '(' (parenthesis) are all tokens of C program.





Types of tokens.

1. Keywords

Examples: float, int, double, while, for.

2. Identifiers

Examples: main, amount

3. Constants

Examples: 12.4, 7894

4. Strings

Examples: "CSM", "Thursday"

5. Special Symbols

Examples: [,], {, }, (,)

6. Operators

Examples: +, *, /



The C character set

The C character set includes the upper case letters A to Z, the lower case a to z, the decimal digits 0 to 9 and certain special characters.

✖ Letters

a, b, c,z

✖ Digits

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

✖ Special characters

~ , . ; : ? ` " ! () [] { } / \ < > = +
- \$ # @ & * % ^

Character/ Symbol	Meaning	Character/ Symbol	Meaning
~	tilde	,	Comma
.	Period	;	Semicolon
?	Ques. mark	:	Colon
"	Double Quote	'	Single Quote
(Left parenthesis)	Right Parenthesis
[Left Bracket]	Right Bracket
{	Left Brace	}	Right Brace
/	Slash	\	Back Slash
<	Less than	>	Greater than
=	Equal to	!	Exclamatory Mark
-	Minus	+	Plus
#	Hash	\$	Dolor Sign
&	Ampersand	*	Asterisk (or star)
%	Percent	^	Carat
	Underscore		Vertical Bar



Identifiers

Identifiers are distinct names given to program elements such as constants, variables, etc.

An Identifier is a sequence of letters, digits, and the special character '_' (underscore).

1. It must start with either a letter or underscore. '_'
2. No commas or blanks are allowed within a variable name.
3. The upper case and lower case letters are treated as distinct, i.e., identifiers are case-sensitive.
4. An identifier can be of any length.
5. No special symbol can be used in a variable name.



Keywords

- ✓ Keywords are predefined tokens in **C**. These are also called reserved words.
- ✓ Key words have special meaning to the C compiler. These key words can be used only for their intended action; they cannot be used for any other purpose.
- ✓ **C** has **32** keywords.



The standard keywords are

auto	break	case	char	const
continue	default	do	double	else
enum	extern	float	for	goto
if	int	long	register	return
short	signed	sizeof	static	struct
switch	typedef	union	unsigned	void
volatile	while			



Data types

- ➡ A data type defines a set of values and the operations that can be performed on them.
- ➡ Every datatype item (constant, variable etc.) in a C program has a datatype associated with it.
- ➡ C also has a special datatype called void, which, indicates that any data type, i.e., no data type, does not describe the data items.



Data types	Description	Size (No. of Bytes)	Range
Char	Single character	1	0 to 255
Int	An integer	2	-32768 to +32767
Float	Floating point number	4	-2,147,483,648 to +2,147,483,647
Double	Floating point number	8	Approximately 15 digits of Precision
Void	No data type	0	
signed char	Character	1	-128 to 127
unsigned char	Unsigned character	1	0 to 255
short signed int	Short signed integer	2	-32768 to +32767
short unsigned int	Short unsigned integer	3	0 to 65535
Long signed int	Long signed integer	4	-2,147,483,648 to +2,147,483,647



Constants and Variables

A constant is a literal, which remain unchanged during the execution of a program.

A constant is a fixed value that cannot be altered during the execution of a program.

C constants can be classified into two categories.

- ✓ Primary Constants
- ✓ Secondary Constants



Rules for constructing Integer constants

- ✗ An integer constant must have at least one digit.
- ✗ It should not contain either a decimal point or exponent.
- ✗ If a constant is positive, it may or may not be preceded by a plus sign. If it is a negative, it must be preceded by a minus sign.
- ✗ Commas, blanks and non-digit characters are not allowed in integer constants.
- ✗ The value of integer constant cannot exceed specified limits.
The valid range is
-32768 to +32767.



Real constants

Real values are often called **floating-point** constants. There are two ways to represent a real constant decimal form and exponential form.

In exponential form of representation, the real constant is represented in two parts. The part appearing before 'e' is called **mantissa**, whereas the part following 'e' is called **exponent**.



Rules for constructing Real constants

- ✗ The mantissa part and the exponential part should be separated by a letter e.
- ✗ The mantissa part may have a positive or negative sign.
- ✗ Default sign of mantissa part is positive.
- ✗ The exponent must have at least one digit, which must be a positive or negative integer. Default sign is positive.
- ✗ Range of real constants expressed in exponential form is - 3.4e38 to 3.4e38.



Rules for constructing Characters constants

- ✗ A character constant is a single alphabet, a single digit or a single special symbol enclosed within single inverted commas. Both the inverted commas point to the left. For example, `'A'` is valid character constant whereas `A` is not.
- ✗ The maximum length of a character constant can be 1 character.

Note: Every character has its **ASCII** (American Standard Code for Information Interchange) value. That means every character is interchange with integer constant. For example, `'A'` value is **65** and `'a'` value is **97**.



String constants

- ✗ A string constant is a sequence of characters enclosed in double quotes. The characters may be letters, numbers, blank space or special characters.
- ✗ Note that "" is **null string** or **empty string**. And the single string constant "A" is not equivalent to the single character constant 'A'.
- ✗ Each string constant must end with a special character '\0'. This character is called **null character** and used to terminate the string. The compiler automatically places a null '\0' character at the end of every string constant.



Escape sequence

Some non-printing characters and some other characters such as double quote ("), single quote ('), question mark (?) and backslash (\), require an escape sequence. A list of commonly used backslash character constants is given below.

Escape Sequence	Meaning	ASCII value	Escape Sequence	Meaning	ASCII value
<code>\a</code>	Bell	7	<code>\r</code>	Carriage return	13
<code>\b</code>	Back Space	8	<code>\"</code>	Double Quote	34
<code>\t</code>	Tab	9	<code>\'</code>	Single Quote	39
<code>\n</code>	New line	10	<code>\?</code>	Question Mark	63
<code>\v</code>	Vertical tab	11	<code>\\</code>	Back Slash	92
<code>\f</code>	Form feed	12	<code>\0</code>	Null	0